Computational Science Technical Note **CSTN-020**

Roles of Rule-Priority Evolution in Animat Models

K. A. Hawick and H. A. James and C. J. Scogings

2005

Evolutionary behaviour in "animat" or physical-agent models has been explored by several researchers, using a number of variations of the genetic algorithmic approach. Most have used a bio-inspired mutation/evolution of low-level behaviours or model properties and this leads to large and mostly "uninteresting" model phase-spaces or fitness landscapes. Instead we consider individual animats that evolve their priorities amongst short-lists of high-level behavioural rules rather than of lower-level individual instructions. This dramatically shrinks the combinatorial size of the fitness landscape and focuses on variations within the "interesting" regime. We describe a simple evolutionary survival experiment, which showed that some rule-priorities are drastically more successful than others. We report on the success of the rule-priority evolutionary approach for our predator-prey animat model and consider how it would apply to more general agent-based models.

Keywords: behaviour rules; priorities; evolution; animat models

BiBTeX reference:

```
@INPROCEEDINGS{CSTN-020,
        author = {K. A. Hawick and H. A. James and C. J. Scogings},
        title = {Roles of Rule-Priority Evolution in Animat Models},
        booktitle = {Proc. Second Australian Conference on Artificial Life (ACAL 2005)},
        year = {2005},
        pages = {99-116},
        address = {Sydney, Australia},
        month = {5-8 December},
        keywords = {behaviour rules; priorities; evolution; animat models}
}
```

Roles of Rule-Priority Evolution in Animat Models

K.A. Hawick, H.A. James and C.J. Scogings
Institute of Information and Mathematical Sciences
Massey University – Albany
Private Bag 102-904 North Shore, Auckland, New Zealand
Email: {k.a.hawick, h.a.james, c.scogings}@massey.ac.nz
Tel: +64 9 414 0800 Fax: +64 9 441 8181

23 June 2005

Abstract

Evolutionary behaviour in "animat" or physical-agent models has been explored by several researchers, using a number of variations of the genetic algorithmic approach. Most have used a bio-inspired mutation/evolution of low-level behaviours or model properties and this leads to large and mostly "uninteresting" model phase-spaces or fitness landscapes. Instead we consider individual animats that evolve their priorities amongst short-lists of high-level behavioural rules rather than of lower-level individual instructions. This dramatically shrinks the combinatorial size of the fitness landscape and focuses on variations within the "interesting" regime. We describe a simple evolutionary survival experiment, which showed that some rule-priorities are drastically more successful than others. We report on the success of the rule-priority evolutionary approach for our predator-prey animat model and consider how it would apply to more general agent-based models.

Keywords: behaviour rules; priorities; evolution; animat models.

1 Introduction

Evolutionary behaviour in physical agent or animat models [13] is both a philosophically intriguing problem and also a computationally demanding one. Other workers have demonstrated exciting emergent [11] properties of animat models, starting from a set of very low-level instructions or microscopic behaviours [1, 5, 10]. It now seems indeed reasonable to describe collections of such animats as artificial life [2, 9, 12] systems. We have previously reported on emergent macroscopic "life-forms" or large scale patterns in a predator-prey model with some carefully chosen microscopic behaviours [4, 6, 7]. Introducing evolutionary behaviour into our model to evolve a better predator or a "longer surviving prey" is interesting but involves what is essentially a brute force exploration of the model's phase space. In this paper we report on what happens when we do **not** evolve **microscopic behaviour** or instructions which are all assumed to have a priori probabilities, but rather what happens when we assume the micro-animat behaviour to be made up of a **pre-selected set** of **high-level rules**. We apply evolutionary algorithms to exploring combinations of pre-evolved rule priorities. We believe this is an important mechanism in real life, in that presumably many of the high level features in real life-forms, once established, can be reused in interesting ways. We consider the analogy with bio-programming and real programming is that we are experimenting with different combinations of relatively small numbers of sub-programs rather than arbitrary large combinations of micro-instructions.

In this paper we briefly summarise and review our predator-prey animat model in section 2. We show some pictorial results of different rule combinations in section 3 and discuss some parametric and statistical metrics to characterise the overall results of simulations and model behaviour in section 4. We report on an evolutionary survival experiment in section 5 and explore how the formulation and application of this hierarchical rule prioritisation approach might be generalised for other animat simulations in section 6.

2 Rule-Based Model

Our model [4,6,7] is based around a set of rule-controlled individual animats which can move, breed, eat and die in a "flatland" of discrete x,y integer coordinates. More than one animat can exist on a single physical cell and the boundaries are not fixed: the size of our model world simply expands as animats explore it. In all our experiments we initialise the model with a block of animats near the origin and time-step the rules so each animat has the opportunity to take one action at each time step.

Two well-known systems with similar objectives to ours are Tierra [10] and Avida [1].

2.1 Tierra and Avida

Within both Tierra and Avida each animat (cell) is represented by a set of integers that represent a command string, consisting of low-level instructions for the individual cell. Each cell has an internal maximum command string length but not all instruction locations may be filled or valid. Instructions are commands such as "nop", "if-not-0", "inc", "dec", "push" and "pop" for the simple accumulator-based cells. They are a kind of simple cell programming language and include instructions to divide cells and attempt to inject instruction sequences into other cells. Cells are contained within a structure called a GeneBank.

One of the fundamental differences between Tierra and Avida is that cells are considered to be 'in a soup' in Tierra, whereas in Avida cells interact only with their (nearest or next-nearest) neighbours on a periodic 2-dimensional grid structure.

Depending on the characteristics of the system under consideration in Tierra and Avida, cells can either undergo reproduction or random mutations. In reproduction the user has control over how the candidate that will reproduce is chosen: either the oldest or largest can be chosen, the one with empty space near it, or a random cell.

Users can choose to enable point mutations of the entire population and mutations on certain operations such as copying cell instruction sets, dividing instruction sets between parent and daughter cells and deletion of instructions from cells. If point mutation is enabled in the system, then at the end of each evolutionary step inside both systems the total population of cells inside the GeneBank is considered as targets for mutation.

Using point mutations, the number of mutations that are to occur is given as the product of the number of active cells inside the GeneBank multiplied by the maximum creature size multiplied by the probability of a point mutation within the system. Once the total number of mutations has been calculated, then random cells are chosen from within the GeneBank, and inside those cells a random int (defining the cell's instruction sequence) is changed, thus introducing a new instruction. Note that it is possible that a single cell may be hit with repeated mutations, making it evolve more quickly than other cells in the system.

Thus, the rules in both these systems are very low level. Furthermore, after the evolution process of a cell has completed, there are no guarantees that its instruction set will produce any meaningful set of operations, let alone a "better" individual.

There are no simple ways, in Avida or Tierra, to "pre-package" a group of low-level instructions into a high-level "routine" for easily manipulation. This is precisely the approach that we take in our exploration of how we can better evolve a predator-prey system: we have defined a small number of operations that we believe characterise the behaviours of our predators and their prey.

2.2 Our Predator-Prey Model

We first developed our model with only two very definite animat types. Predators (known as "foxes") and prey (known as "rabbits") coexist in the "flatland" simulated by the model.

Within our predator-prey model there are several redundant parameters that control microscopic details of the animats. These rules fall into two broad categories: (a) rules affecting the environment such as how far a rabbit can "see" or how long a fox takes to get hungry; and (b) the set of rules that govern the behaviour of an individual animat. We describe choices for category-a parameters in previous work, but generally most of the model behaviours are insensitive to these. In this paper we focus on permutations of category-b rules that govern behaviour of individual animats. These rules are typical of the type of rule that is changed in a system with animats that are evolving by following a genetic algorithm such as Tierra or Avida.

We are exploring how the model varies by changing the rule priority permutation that animats use, rather than changing the basis set of possible rules themselves.

In our previous work, we have used the same set of rules for every animat. These rules were chosen by us as a good base set providing a rich set of pattern formation and are stable against small variations. An outline of these rules is:

A fox will:		A rabbit will:	
0	eat a rabbit if adjacent;	0	move away from an adjacent fox;
1	move towards a visible rabbit if hungry;	1	breed if adjacent to another rabbit;
2	breed if adjacent to another fox;	2	move towards another visible rabbit;
3	move towards a visible fox if not hungry;	3	or move randomly.
4	or move randomly.		

The order of these rules is important. Each rule has a condition and if that condition is true, the rule is executed and any later rules will be ignored. Thus rule 0 has a higher priority than rule 1 and so forth. For example, if a fox is not hungry and is adjacent to another fox but not adjacent to a rabbit, than rule 2 (breeding) will be executed and rules 3 and 4 will be ignored. Note that the conditions for rules 0 and 1 will be false in this case. The "move randomly" rule is a catch-all and does not have a condition. Thus if all else fails the animat will move randomly.

We can therefore denote the behaviour of a particular animat i at time t by $\mathcal{B}_i(t)$ which will result in the action expressed by the rule priorities in effect.

Suppose the animat i has a current rule list $\mathcal{R}_i = [r_0, r_1, r_2, ..., r_j]; j = 0, 1, ..., N_R - 1$ Rules are evaluated in strict order and the first that can be applied is actioned. In this paper we restrict the list length N_L to being identical to the number of possible rules N_R and so each of our rules appears precisely once in an animat's list. Our rules are formulated as a "first matching" so that it is not useful to duplicate rules in the list, but it would be possible to omit some possible rules randomly or otherwise, so that $N_L <= N_R$.

A permutation $\mathcal{P}_k; k = 0, 1, ..., N_P - 1$ such as $\mathcal{P}_k = [1, 0, 2, 3, 4]$ can be applied to the rule priority lists. It is the space of possible permutations $\mathcal{P}_k$ that we are exploring in this present paper. The number of possible permutations is drastically smaller than the number of combinations of an arbitrary string of N_R instructions $N_R^{N_R}$. For operational reasons the last rule is always the "move randomly" rule in our present experiments, so we can only in fact permute only $4 = N_R - 1$ rules for the fox model described. The number of permutations is therefore $4! = 24$.

3 Resultant Behaviours from Prioritisation

We have investigated the macroscopic pattern formations that result from using different permutations of rules. In this paper we focus on the fox-species and the resulting behaviours for the 24 different subspecies of fox in the presence of "normal" or base species rabbits. We present some configuration snapshots showing the macroscopic pattern variations and summarise our findings in table 1 for each of the 24 fox subspecies.

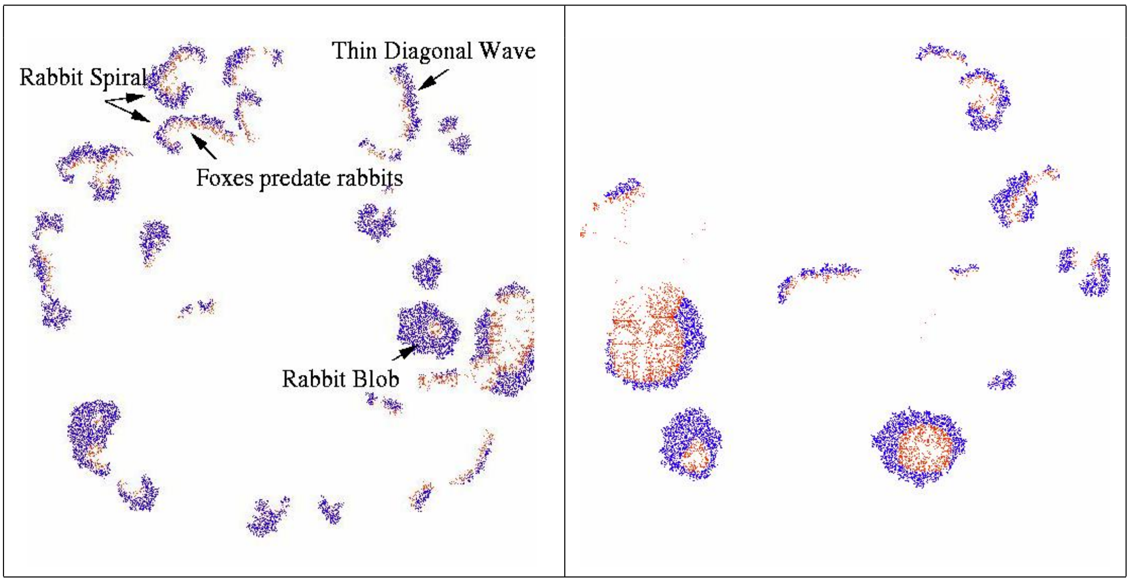

Figure 1: Situation at step 800 for (a) the original Fox rule set 0-1-2-3-4 and (b) Fox rule set 2-0-3-1-4

The following figures show the situation in the model at time step 800. We chose this time regime, as our previous work suggests that all system memory of the initial animat configuration is erased by step 400. Each run of the model uses the same random number seed so the outcome would have been identical each time except that the order of the fox rules has been changed to one of the 24 possible permutations.

A number of experiments were run where the order (and thus the priorities) of the rules were changed. In order to limit the number of experiments, one species (either rabbit or fox) retained its usual rules while the order of the rules for the other species were changed. Each rule set is referred to by a sequence of digits which show the new order for the rules. For example: rule set 1-3-0-2-4 means that the rabbit rules are as usual and the fox rules are in the order 1, 3, 0, 2 and 4. In this paper we focus on fox rule combinations, but where a rule set is given if it has 5 digits it refers to a fox rule, if it has 4 digits it is a rabbit rule.

In previous work we analysed the macroscopic patterns that emerge from the microscopic model [8]. We identified shapes we refer to as: rabbit clumps; defensive lines; horizontal, vertical and diagonal wave-fronts and left or right spirals. We use this vocabulary in discussing and analysing the configuration snapshots shown below.

Figure 1 (a) shows step 800 for the predator-prey model with the original (base) rule set and (b) shows time step 800 for the fox rule set 2-0-3-1-4. These two diagrams illustrate the two main visual effects caused by changes in rule priorities. Certain rule sets (including the original set 0-1-2-3-4) produce an overall "circular" pattern with clusters of animats spread more or less evenly across the area. These clusters comprise the usual spirals, wave fronts and dense "blobs" previously analysed in [4]. Each wave front consists of a leading edge of rabbits fleeing from a following line of foxes. It is noticeable that the wave fronts in this type of diagram are predominantly "compact", that is the line of foxes is keeping up with the rabbits.

The second diagram in Figure 1 (rule set 2-0-3-1-4) illustrates the second type of effect where the overall circle pattern is destroyed and larger distinct clumps form. In addition many clumps show that foxes have scattered away from the line of rabbits. This effect is caused by rule 1 (move towards rabbit) appearing low down in the list of priorities.

Both the diagrams in Figure 1 contain relatively low numbers of animats - 12,609 rabbits and 2,683 foxes with the original rule set and 8,004 rabbits and 3,648 foxes in rule set 2-0-3-1-4. Note that the number of animats in a "good circle" pattern is always significantly higher than in the "degraded circle" where the lower numbers of animats is caused by rule 3 (move towards another fox) appearing low down in the priority list. This shows how rule 3 has more of an effect on the population than rule 2 (breed) which can only be used if another fox is adjacent.

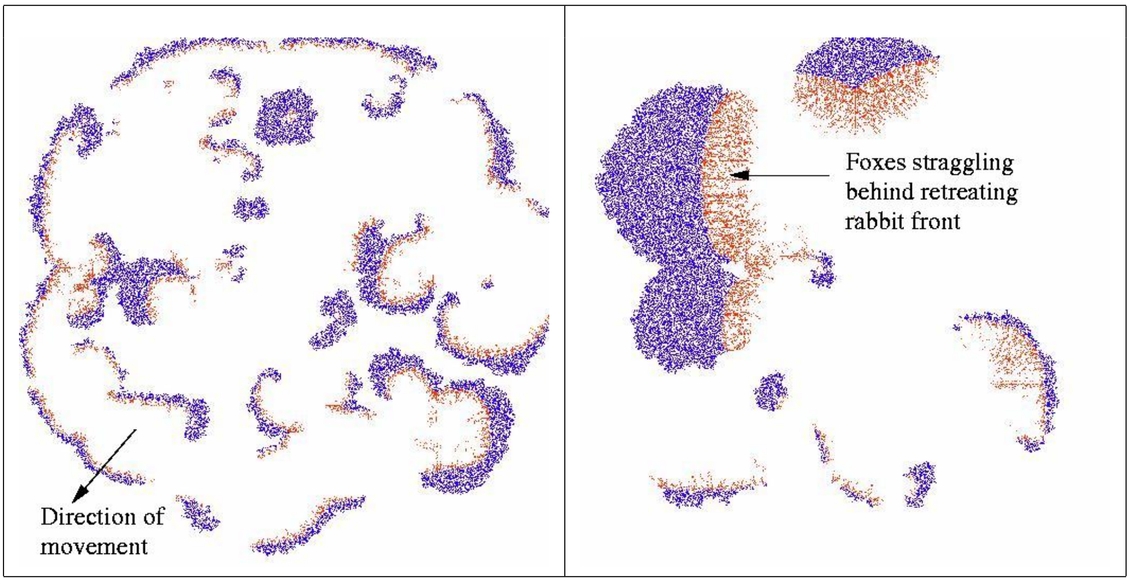

Figure 2: Situation at step 800 for (a) Fox rule set 3-0-1-2-4 and (b) Fox rule set 0-3-2-1-4. In contrast to figure (b), figure (a) has a good circular shape.

Figure 2 shows two rules sets which illustrate an "in-between" situation. Rule set 3-0-1-2-4 contains a reasonable circle but several wave fronts show foxes falling slightly back due to rule 1 being in the centre of the list of priorities. Whereas rule set 0-3-2-1-4 shows a poor circle with distinct foxes trailing behind the rabbit lines due to rule 1 being at the end of the list. Both rule sets contain relatively high levels of animats due to the high priority given to rule 3 – 21,277 rabbits and 6,139 foxes in 3-0-1-2-4 and 22,129 rabbits and 8,021 foxes in 0-3-2-1-4.

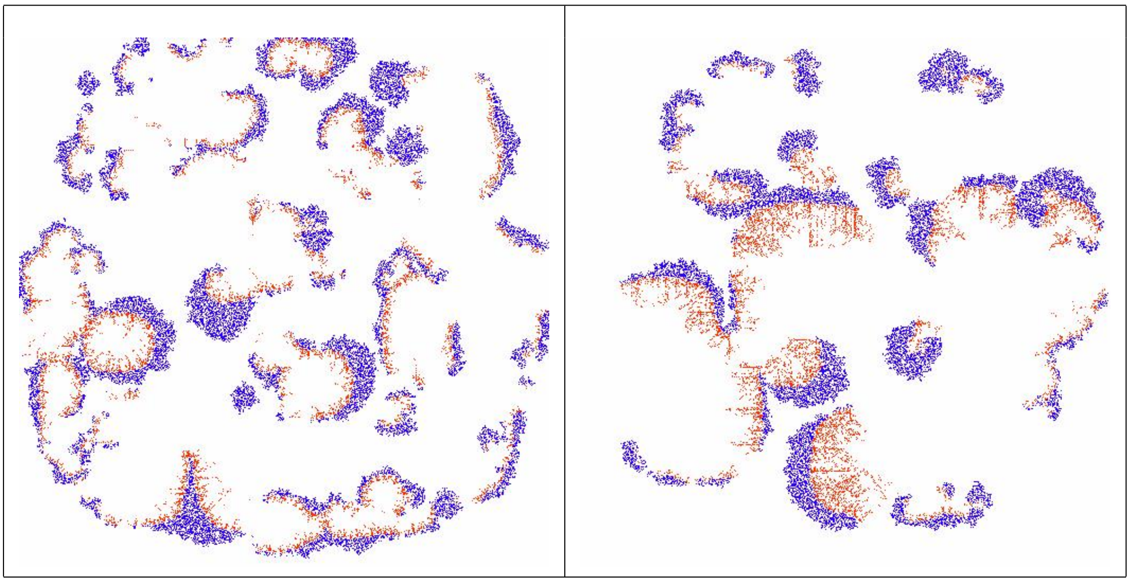

Figure 3: Situation at step 800 for (a) Fox rule set 3-1-2-0-4 and (b) Fox rule set 3-0-2-1-4

In Figure 3, rule set 3-1-2-0-4 contains a high number of animats – 23,735 rabbits and 8,818 foxes. Likewise, rule set 3-0-2-1-4 also contains high numbers of animats (for a diagram with a degraded circle) with 17,429 rabbits and 8,974 foxes. The degraded circle and scattered foxes in 3-0-2-1-4 is due, as usual, to rule 1 being low in the list. The relatively high number of animats in both these rule sets is due to rule 3 being the first rule in the list.

Figure 4 shows results for foxes following their original rules while the rabbit rule order has been changed. Rabbit rule set 2-0-1-3 shows low numbers of rabbits (5,790) because rule 1 (breed) has moved to the end of the list just before rule 3. Thus less rabbits breed and the fox population is also low (1,329) because of a lack of prey. Rule set 0-2-1-3 shows rabbits "herding" together because of the combination of rule 0 (move away from fox) and rule 2 (move towards other rabbit) as the first two rules in the list. This causes many rabbits to have a "double movement" and thus large clusters of rabbits form in front of pursuing foxes.

In general the rule list permutations we have examined are remarkably robust and all give rise to macroscopic patterns without extreme population explosions or extinctions. Generally the length scales of the model are fairly stable too – all the configurations shown in figures 1 to 4 are all on the same length scale.

Table 1 contains all tested rule permutations for foxes and table 2 contains all tested rule permutations for rabbits. Each entry in these tables indicates the number of foxes and rabbits at step 800.

Several of the rule sets were found to be degenerate. In particular, if rule 4 (random move) for foxes precedes the rules for breeding or eating, then every fox moves randomly and never breeds or eats which results in the fox population becoming extinct within a few time steps. Thus the rule sets of interest always place rule 4 as the last rule in the list. Similarly, rule sets involving the changing of the order of the rules for rabbits always retain rule 3 as the last in the list.

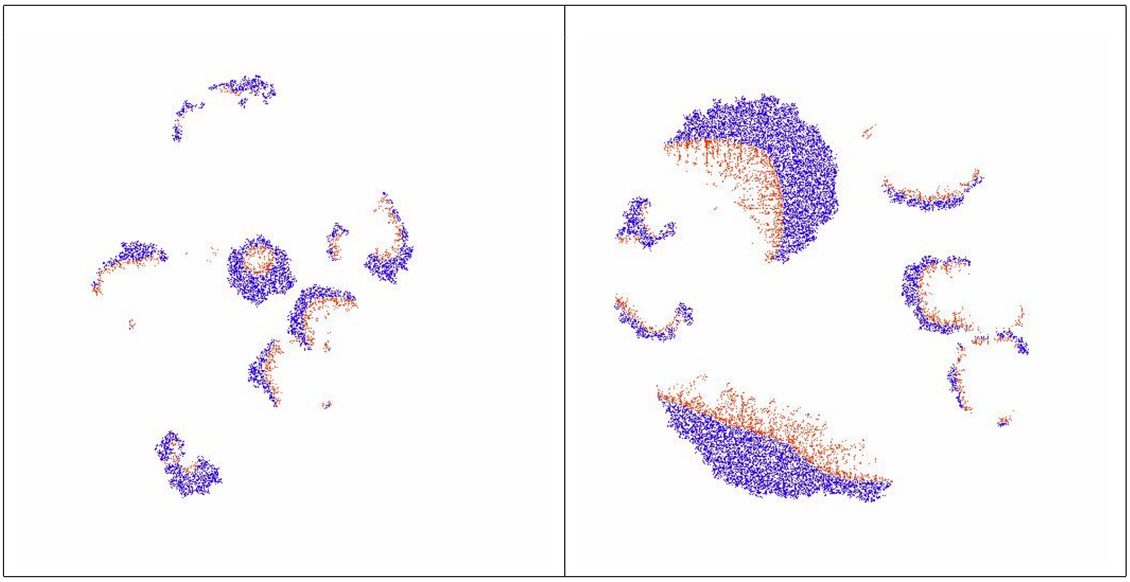

Figure 4: Situation at step 800 for (a) Rabbit rule set 2-0-1-3 and (b) Rabbit rule set 0-2-1-3

Permute code	Priority List	Rabbits at step 800	Foxes at step 800	Description
0	0-1-2-3-4	12,609	2,683	good circle, no straggling (original rule set)
1	0-1-3-2-4	21,598	8,113	good circle, no straggling
2	0-2-1-3-4	3,800	2,557	no circle, some straggling
3	0-2-3-1-4	3,800	2,557	identical to 0-2-1-3-4 (permute code 2)
4	0-3-1-2-4	21,598	8,113	identical to 0-1-3-2-4 (permute code 1)
5	0-3-2-1-4	22,129	8,021	no circle, heavy straggling
6	1-0-2-3-4	16,217	5,163	good circle, no straggling
7	1-0-3-2-4	23,715	7,644	good circle, no straggling
8	1-2-0-3-4	15,052	6,784	weak circle, slight straggling
9	1-2-3-0-4	10,884	5,534	good circle, no straggling
10	1-3-0-2-4	24,180	6,881	good circle, slight straggling
11	1-3-2-0-4	23,735	8,818	good circle, no straggling
12	2-0-1-3-4	8,004	3,648	no circle, heavy straggling
13	2-0-3-1-4	8,004	3,648	identical to 2-0-1-3-4 (permute code 12)
14	2-1-0-3-4	15,471	7,574	no circle, blobs, heavy straggling
15	2-1-3-0-4	11,131	10,961	weak circle, some straggling
16	2-3-0-1-4	6,172	7,192	weak circle, heavy straggling
17	2-3-1-0-4	11,131	10,961	identical to 2-1-3-0-4 (permute code 15)
18	3-0-1-2-4	21,277	6,139	good circle, no straggling
19	3-0-2-1-4	17,429	8,974	weak circle, heavy straggling
20	3-1-0-2-4	24,180	6,881	identical to 1-3-0-2-4 (permute code 10)
21	3-1-2-0-4	23,735	8,818	identical to 1-3-2-0-4 (permute code 11)
22	3-2-0-1-4	10,723	8,100	weak circle, some straggling
23	3-2-1-0-4	13,669	5,774	weak circle, some straggling

Table 1: $4! = 24$ permutations of rules controlling fox behaviour showing numbers of rabbits and foxes at step 800 and some description of the observed behavioural effects

Of the useful rule sets for foxes, six were found to be identical to other rule sets. This occurs when rule 3 and rule 1 follow each other since rule 1 is only used if the fox is hungry and rule 3 is only used if the fox is not hungry thus only one of these two rules will ever be used. If they appear in sequence they have the same effect as if they appeared in the opposite order – thus, for example, rule set 0-1-3-2-4 has an identical effect to rule set 0-3-1-2-4.

Permute code	Priority List	Rabbits at step 800	Foxes at step 800	Description
0	0-1-2-3	12,609	2,683	good circle, no straggling (original rule set)
1	0-2-1-3	14,831	4,522	no circle, huge rabbit herds, heavy straggling
2	1-0-2-3	18,474	9,115	good circle, slight straggling
3	1-2-0-3	14,461	4,938	good circle, no straggling
4	2-0-1-3	5,790	1,329	no circle, no straggling
5	2-1-0-3	9,018	3,500	weak circle, no straggling

Table 2: 3! = 6 permutations of rules controlling rabbit behaviour showing numbers of rabbits and foxes at step 800 and some description of the observed behavioural effects

The frequency that animats (of a particular type) will enact a particular rule can be measured for a particular time-slice and averaged over all animats of that type to obtain what we refer to as action-probabilities where $P_j(t)$ is the probability that action or rule j is enacted at time t. In fact, we expect the dependency to be more complicated as our system is known to have a periodically varying population envelope function for animats of a particular type (e.g. foxes)

The measured action-probabilities are both time dependent and also depend on the circumstances (local environment) of individual animats. However, averaging over all similar animats (eg “foxes”) in the population reveals that the mean probability of rule j is a good differentiator between the different behaviours that arise from different rule permutations.

The rule action-probabilities were measured during several runs of the model with different rules sets and appear in table 3.

Rule	original set	set 0-2-1-3-4	set 2-1-0-3-4	set 3-0-2-1-4
0	0.081	0.052	0.042	0.048
1	0.432	0.495	0.527	0.483
2	0.029	0.035	0.039	0.034
3	0.074	0.071	0.067	0.085
4	0.385	0.346	0.326	0.350

Table 3: Rule priority action-probabilities for various rule sets from step 1 to step 1000

Table 3 clearly shows that the two seemingly most important rules, namely rule 0 (eat) and rule 2 (breed) are actually used the least. This is because they depend heavily on their accompanying “precondition” rules 1 (move towards a rabbit) and 3 (move towards another fox). In fact, foxes spend roughly half their time moving towards food (rule 1) and this is consistent even when rule 1 is given a low priority (see 3-0-2-1-4). Similarly, rule 3 is also used in a consistent manner although there is a slight increase in use when it is given a high priority (see 3-0-2-1-4).

4 Behavioural Metrics and Analysis

As table 1 shows, the different rule priority lists give rise to significant behaviour variations. Some interesting metrics that can be used to characterise these differences are discussed below.

One metric is the size (and density) of typical animat clusters such as wave fronts. It has already been shown how the placement of fox rule 1 (move towards rabbit) can cause foxes to "straggle" behind wave fronts. This appears to be independent of the direction of travel of wavefronts.

	low number of rabbits	medium number of rabbits	high number of rabbits
good circle		0-1-2-3-4, 1-0-2-3-4, 1-2-0-3-4, 1-2-3-0-4	0-1-3-2-4, 1-0-3-2-4, 1-3-0-2-4, 1-3-2-0-4, 3-0-1-2-4
no/weak circle	0-2-3-1-4, 2-0-3-1-4, 2-3-0-1-4	2-1-0-3-4, 2-3-1-0-4 3-2-0-1-4, 3-2-1-0-4	0-3-2-1-4, 3-0-2-1-4

Table 4: Rule sets grouped according to number of rabbits at step 800

Another metric is the relative animat populations. Table 4 shows how changes to the priority of the rules for foxes affect the rabbit population. Although the grouping criteria are vague, it is clear that rule 0 (eat) has little or no effect. The rabbit population size is governed by the placement of fox rule 2 (breed). If rule 2 is placed near the top of the list, then more foxes will be produced to prey on the rabbit population which will consequently be lower. Similarly, if rule 2 is placed near the end of the list then a higher rabbit population results. The rule priority also affects the general dispersal of the population. If rule 1 (move towards rabbit) is placed near the top of the list then a good circular pattern will result. This is because foxes are often moving towards rabbits which, in turn, flee from the approaching foxes, forming more wave fronts in an expanding circular formation.

	low number of foxes	medium number of foxes	high number of foxes
good circle	0-1-2-3-4, 1-0-2-3-4, 1-2-3-0-4	0-1-3-2-4, 1-0-3-2-4, 1-3-0-2-4, 3-0-1-2-4	1-3-2-0-4
no/weak circle	0-2-1-3-4, 2-0-1-3-4	0-3-2-1-4, 1-2-0-3-4, 2-1-0-3-4, 2-3-0-1-4, 3-2-0-1-4, 3-2-1-0-4	2-1-3-0-4, 3-0-2-1-4

Table 5: Rule sets grouped according to number of foxes at step 800

Table 5 shows how changes to the priority of the rules for foxes affect the fox population. The grouping is not as clear as that for the rabbit population but does show certain trends. Once again the priority given to rule 0 (eat) seems to have no effect.

Another metric is an analysis of fox lifestyle and breeding habits which change with each rule set. For example the mean lifetime of foxes using the original rule set 0-1-2-3-4 is 32 (time steps) whereas the mean lifetime of foxes using rule set 2-0-1-3-4 is 26. In the original rule set 40% of foxes die of old age (at 50 timesteps) and 60% starve to death whereas in rule set 2-0-1-3-4, 20% die of old age and 80% starve to death. These differences occur because the original rule set leads with rule 0 (eat) and 2-0-1-3-4 leads with rule 2 (breed).

5 An Evolutionary Survival Experiment

We believe we have identified the gross behaviour classes of the possible rule set permutations in our model. We can therefore introduce a very simple genetic evolution process into the model and observe which subspecies is most fit. Fitness of course can have several interpretations. In our present discussion we simply take it to mean survivability. We set up a model run whereby all subspecies of the model configuration are substituted randomly by a particular subspecies. In the case of foxes, the relative populations of each subspecies rapidly diverge.

5.1 Evolution Procedure

In this simplified evolutionary process we simply inject the different subspecies into the fox population after the initial set up. Individual pairs of animats breed producing offspring that are randomly (50:50) the same type as one or either parent. We have not yet introduced any mutation. Unsuccessful species therefore die out and are lost from our population. That is the aim of this particular experiment – simply to determine which are the most likely subspecies to survive.

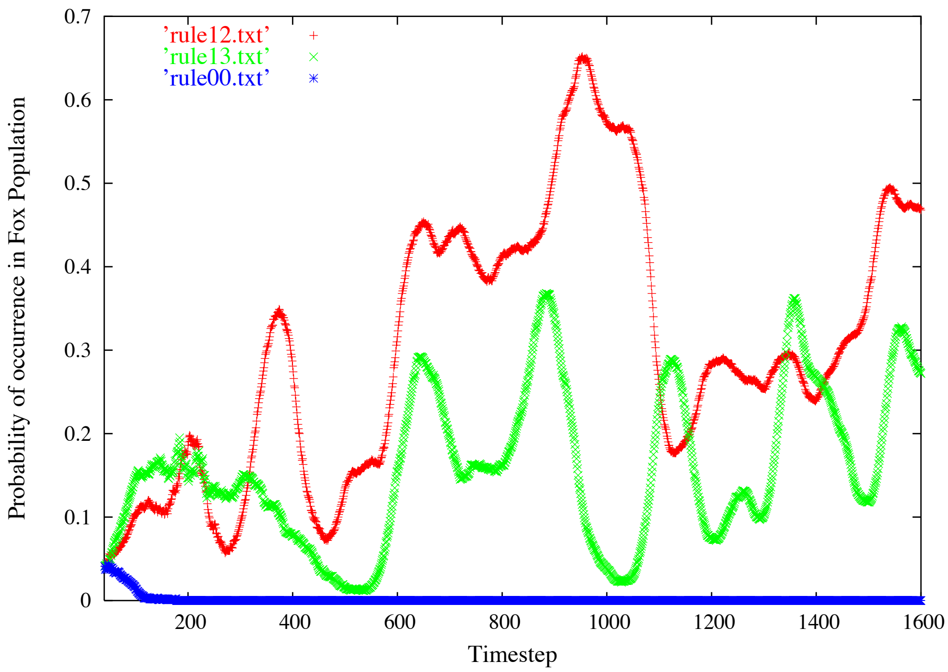

Figure 5: Survival frequencies for the different species of Foxes

We have tried several random seed combinations for this experiment and the emergent pattern is similar in all cases. We conclude that there are enough random effects in this experiment to separate out pathological spatial effects. We anticipate that certain conditions that arise for some pockets of the population such as being at the end of a spiral or being at the heart of a rabbit clump might lead to special behaviour and affect survivability. In this paper we only report on behaviour averaged over the whole system and hence without analysis of potential anomalies.

We believe this is a useful exercise to explore survivability in our model, prior to the more open ended studies involving genetic cross over and mutation that will considerably expand the size of the fitness landscape.

5.2 Survivability

Figure 5 shows how the relative frequencies of the different subspecies of fox dramatically diverge from an initial equality. Foxes with a rule set starting with rule 2 (breed) have a very much stronger tendency to survive.

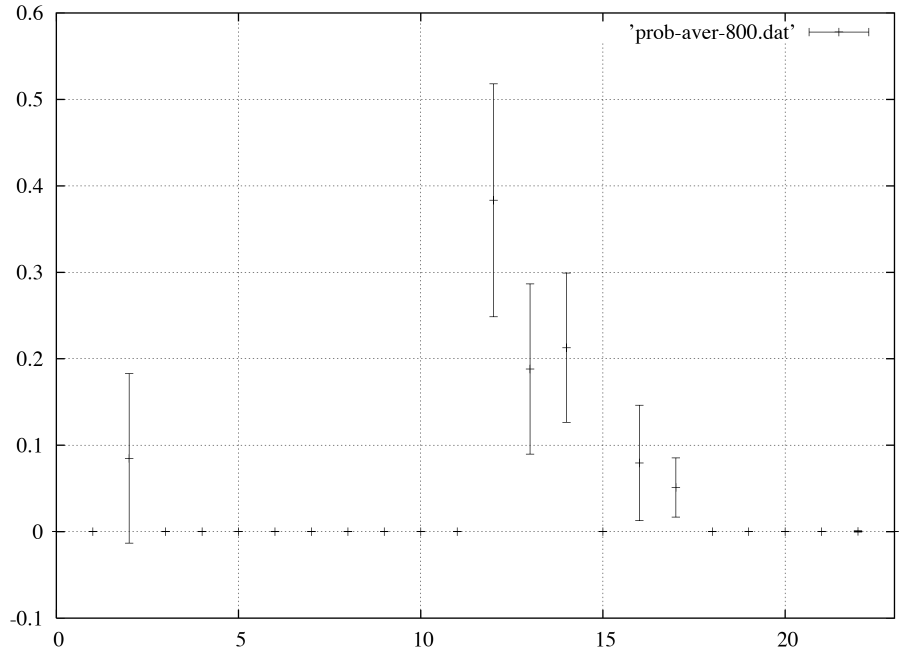

Figure 6: Survival frequencies and variances for all fox types averaged over last 800 steps in a 1600 step run where multiple species were injected at step 40

Figure 5 shows the measured survival frequencies for fox subspecies, normalised over the entire remaining population. The curves for rules 0, 12 and 13 are shown for illustrative purposes. Generally as expected the population follows periodic boom-bust variations. The interesting feature that arises from the different fox subspecies is due to their different relative propensities for survival. Generally rules fall into one of two survival categories - long term survival or short term extinction. In this particular experiment there is no evolutionary mechanism for an extinct subspecies of fox to be later re-introduced into the population.

Figure 6 shows the relative survival frequencies s of the sub type populations averaged over the last 800 steps of a 1600 step run where the different subspecies were injected at step 40. Prior to step 40 the run used the original rule set for all foxes just to build up a large and uncorrelated base population. The graph shows the value and variance of the relative survival frequencies. Note that most subspecies are effectively extinct and the subspecies that have survived are rules 12,13,14,16,17 which all have rule 2 (breed) as their top priority. However anomalously foxes of rule set 2 also survive and rule set 15 which also has (breed) as its top priority does not. This latter may be explained by the low priority rule 15 (2-1-3-0-4) foxes give to eating – therefore they easily starve. Rule set 2 (0-2-1-3-4) survives because of lucky chance combinations of eating and breeding (and their precondition rules for moving towards food and breeding partners respectively) in the priority list.

6 Generalising the Approach

By considering high-level "common sense" ideas for the behaviour of individual animats, we have arrived at a permuted-rules model that is amenable to evolution. In this paper we have only considered very simple probabilistic inheritance. It is relatively straightforward to apply point mutation into the animat rule lists, but it is not entirely clear yet what the best way to provide genetic crossover is.

It is interesting to ask what makes a successful animat anyway? Survival has meaning in the context of particular subtypes. Our model is simple enough that we can control and limit some of the interactions that would otherwise lead to an irreducible set of species-species reactions. It is likely however that we can engineer animats with particular traits that will be especially "fit" for certain circumstances that may occur in the model.

We have managed to introduce a very limited sort of evolutionary operation into our predator prey model. The evolution operator is not very satisfactory albeit adequate for these initial investigations of relative survivability. We believe it is straightforward to introduce point mutations into the process but are still investigating good ways to define a genetic cross over representation and operator that will allow a controlled exploration of new sorts of animat.

There are of course other phenomena to explore. We are currently exploring what might make a particular animat more or less attractive to a potential breeding mate to encourage promotion of particular traits in the animats.

We are also considering the particular local circumstances that arise in the macroscopic patterns that emerge in our model. It is not clear how to link microscopic and macroscopic behaviours, but it is likely that for example foxes at the horns of a wavefront will behave differently from those gorging in the midst of a rabbit clump. We would like to investigate particular survival niches that might arise from such macroscopic effects.

We believe we have considerably reduced the search space for evolution in our model using the permutation approach and fixed rule sets. We believe the rule sets have some relevance to the exploration of gene functions in real life systems.

7 Conclusions

We have codified the important behaviours of our predator prey animats and shown how they can be represented by permutable gene sequences. We have categorised the major properties of fox and rabbit sub-species in our model using just 5 and 4 gene sequences respectively. Our model is remarkably robust and we have been able to identify some key behaviour regimes of stability.

We have a number of developments planned for the future [3]. Firstly, a “do nothing” rule could be added to our basis set and this could be used as the last rule, giving an N_L of 5 and a permutation space of $5! = 120$. We believe such a “nop” rule will only change the time constants in our model but we have yet to investigate this.

Secondly, we have focused on foxes in this paper since they have a richer rule permutation possibilities. We have yet to explore the survival propensities of different types of foxes in the presence of different rabbits. We hope to map out this model space semi-automatically based on the categorisation ideas presented in the present paper.

Finally, we believe our approach makes an exploration of a model phase space much more manageable than the common “shot gun” approach to genetic algorithms. Our approach of combining a carefully controlled evolutionary exploration of model phase space with a microscopic model known to exhibit emergent macroscopic properties may be of relevance to studying other life systems – artificial or otherwise.

Acknowledgements

This work has benefited from the use of Massey University’s “Monte” compute cluster and thanks also to the Allan Wilson centre for Molecular Biology for the use of the “Helix” Supercomputer.

References

[1] Adami, C. Avida (Digital Life Laboratory) Available at http://dllab.caltech.edu/avida

[2] Caltech Digital Life Lab. Web pages available at http://dllab.caltech.edu

[3] Hawick, K.A., James, H.A. and Scogings, C.J. Web site, containing model Snapshots and Movie sequences: *http://www.massey.ac.nz/˜kahawick/alife* Massey University 2004-5

[4] Hawick, K.A., Scogings, C.J. and James, H.A. Defensive Spiral Emergence in a Predator-Prey Model in Proc. Complexity 2004, Cairns, Australia, December 2004.

[5] Holland, J. ECHO Available from http://www.santafe.edu/projects/echo/echo.html

[6] James, H.A., Scogings, C.J. and Hawick, K.A. A Framework and Simulation Engine for Studying Artificial Life in Research Letters in the Information and Mathematical Sciences, Vol 6, May 2004, pp143–155, ISSN 1175-2777. Available from http://iims.massey.ac.nz/research/letters/-volume6/

[7] James, H.A., Scogings, C.J. and Hawick, K.A. Parallel Synchronisation issues in Simulating Artificial Life in Proc. 16th IASTED Int. Conf. on Parallel and Distributed Computing and Systems (PDCS), pp, 815–820, Boston, November 2004.

[8] Hawick, K.A., James, H.A. and Scogings, C.J. A Zoology of Emergent Patterns in a Predator-Prey Model Computational Science Technical Note CSTN-015, Massey University , March 2005.

[9] Levy, S. “Artificial Life” Penguin Books, 1992. ISBN 0-14-023105-6

[10] Ray, T. Tierra Available at http://www.isd.adr.co.jp/˜ray/tierra

[11] Ronald, E.M.A., Sipper, M. and Capcarrère, M.S. Testing for Emergence in Artificial Life, In Advances in Artificial Life: Proc. 5th European Conference on Artifical Life (ECAL'99), Switzerland, 1999 Ed. Dario Floreano and Jean-Daniel Nicoud and Francesco Mondada, Pages 13-20, Pub Springer-Verlag ISBN 3-540-66452-1.

[12] Santa Fe Institute. Available at http://www.santafe.edu

[13] Wilson, S. The Animat Path to AI in *From Animals to Animats 1: Proceedings of The First International Conference on Simulation of Adaptive Behavior*, (pp. 15-21); Meyer, J-A. & Wilson, S. (eds), Cambridge, MA: The MIT Press/Bradford Books (1991).